Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

Memetic micro-genetic algorithms for cancer data classification

Matías Gabriel Rojas[a,*], Ana Carolina Olivera[a,c], Jessica Andrea Carballido[b], Pablo Javier Vidal[a,c]

[a] Instituto Universitario para las Tecnologías de la Información y las Comunicaciones, Consejo Nacional de Investigaciones Científicas y Técnicas, Universidad Nacional de Cuyo, Padre Jorge Contreras 1300, Mendoza, M5502JMS, Mendoza, Argentina

[b] Instituto de Ciencias e Ingeniería de la Computación, Consejo Nacional de Investigaciones Científicas y Técnicas, Universidad Nacional del Sur, San Andres 800, Bahía Blanca, 8000, Buenos Aires, Argentina

[c] Facultad de Ingeniería, Universidad Nacional de Cuyo, Centro Universitario, Mendoza, M5502JMA, Mendoza, Argentina

ARTICLE INFO

Keywords:
Memetic algorithms
Cancer microarray data
Micro-genetic algorithm
Feature selection

ABSTRACT

Fast and precise medical diagnosis of human cancer is crucial for treatment decisions. Gene selection consists of identifying a set of informative genes from microarray data to allow high predictive accuracy in human cancer classification. This task is a combinatorial search problem, and optimisation methods can be applied for its resolution. In this paper, two memetic micro-genetic algorithms (MμV1 and MμV2) with different hybridisation approaches are proposed for feature selection of cancer microarray data. Seven gene expression datasets are used for experimentation. The comparison with stochastic state-of-the-art optimisation techniques concludes that problem-dependent local search methods combined with micro-genetic algorithms improve feature selection of cancer microarray data.

1. Introduction

Classification systems for medical diagnosis are becoming more and more popular in the last decade. They consist of data mining processes that divide data into classes to facilitate diagnosis of different pathologies (Tarle et al., 2019). Cancer disease is one of the most critical areas of research in the medical field. DNA microarray technology allows the analysis of thousands of genes' expression level, which is specifically useful for cancer diagnosis. The capability to predict normal tissue and different tumour types is crucial to patient prognosis and early treatment of disease.

The determination of which genes are useful to predict between possible classes is a difficult task. Irrelevant and redundant genes are not valuable for classification, and moreover, they can also make the analysis harder and more prone to errors (Alonso-Betanzos et al., 2019). As such, effective gene selection methods for cancer are critically necessary. In this context, a feature selection process aims to choose the minimum number of informative genes most predictive for increasing the accuracy in the classification process.

Feature selection is considered to belong to the class of NP-hard combinatorial optimisation problems (Venkatesh & Anuradha, 2019). Feature selection of microarray cancer data consists of taking a subset of n genes from an immense set of N genes to use in the definition of a mathematical model (Narendra & Fukunaga, 1977). These selected features can be viewed as a subset of features that needs to be evaluated as a whole (Xue et al., 2016). For a set of N genes, there are 2^N subsets (Siedlecki & Sklansky, 1993). Traditional deterministic search methods can be computationally costly to find the optimal solution (Dash & Liu, 1997, Liu & Zhao, 2009). To reduce the time complexity, alternative methods have been proposed: greedy techniques (Mao & Tsang, 2013, Min et al., 2014), metaheuristics (Li et al., 2013, Dussaut et al., 2018, Shukla et al., 2020), hybridisation of

[*] Corresponding author.
E-mail addresses: mrojas@mendoza-conicet.gob.ar (M.G. Rojas), acolivera@conicet.gov.ar (A.C. Olivera), jac@cs.uns.edu.ar (J.A. Carballido), pjvidal@conicet.gov.ar (P.J. Vidal).

https://doi.org/10.1016/j.iswa.2022.200173
Received 25 May 2022; Received in revised form 19 August 2022; Accepted 28 December 2022
Available online 3 January 2023

Nomenclature

α	Importance of the accuracy in the fitness function
N	Number of Genes
NP-Hard	Kind of problems without a polynomial solution
μGA	Micro-Genetic Algorithm
MμGA	Memetic Micro-Genetic Algorithm
M	Number of samples from patients
x_i	State of a given gene (selected or non-selected)
k	Number of groups for K-fold Cross-Validation
K-NN	K-Nearest Neighbours Classifier
SVM	Support Vector Machine Classifier
DT	Decision Tree Classifier
FS	Feature Selection
GA	Genetic Algorithm
MA	Memetic Algorithms
PSO	Particle Swarm Optimisation
CGA	Cellular Genetic Algorithm
HC	Hill Climbing Algorithm
p or $popSize$	Population size
$offspringPop$	Offspring population size
V	Offspring population size
$matingPop$	Mating population
P	Population of solutions
MμV1	Memetic Micro-Genetic Algorithm Version 1
MμV2	Memetic Micro-Genetic Algorithm Version 2
U	Number of evaluations for the local search operator
ALL	Acute Lymphoblastic Leukaemia
AML	Acute Myelogenous Leukaemia
ADCA	Adenocarcinoma
MLP	Malignant Pleural Mesothelioma
SRBCT	Small-Blue Round Cell Tumour
EWS	Ewing Sarcoma
BL	Burkitt Lymphoma
NB	Neuroblastoma
RMS	Rhabdomyosarcoma
RBF	Radial Basis Function kernel
BPSO	Binary Particle Swarm Optimisation
VNS	Variable Neighbourhood Search
BTS	Binary Tournament Selection
HUX	Half-Uniform Crossover
BDE	Binary Differential Evolution
PeSOA-C	Penguin Search Optimisation Algorithm with Rapid Convergence
IG-μGA	Information Gain and Micro-Genetic Algorithm
C-HMOSHSSA	Multi-Objective Spotted Hyena Optimiser, hybridised with the Salp Swarm Optimisation
ACO-S	Ant Colony Optimisation-Selection

metaheuristics (Jona & Nagaveni, 2014), and recently, memetic algorithms (Lee & Kim, 2015).

Moreover, memetic algorithms have gained importance due to their ability to combine techniques such as evolutionary algorithms (that are suitable for feature selection -their population-based mechanism produce multiple solutions in a single run-) (Xue et al., 2016) and local search that allows exploiting areas incapable of reaching by canonical techniques (Abu Zaher et al., 2019, Lee & Kim, 2015, Rojas et al., 2020, Yang et al., 2008).

Micro Genetic Algorithm (μGA) is a kind of metaheuristic that uses genetic operators to improve a small population of solutions for search intensification and diversification. To avoid the stagnation of μGA, it uses a restart population method to keep the best solution found and reset the others. The aim of this algorithm is to achieve the most-fit solution by applying high levels of elitism and, at the time, reduce the computational resource usage. However, some problems require the algorithm to work with different operators to improve the general method and incorporate knowledge of the problem. Therefore, the design of a memetic μGA that allows collaboration with another technique and the use of problem-oriented operators could provide robust and numerically effective behaviour (in terms of the classification values obtained).

In this work, two novel Memetic micro-Genetic Algorithms (MμGAs) are introduced to improve the qualities of the canonical μGA in selecting the most relevant genes of microarray cancer data and, thus, increase the classification capability. The main contributions of this paper can be resumed as follows:

- Two novel memetic micro-genetic algorithms (MμGAs) are introduced for feature selection.
  - The first approach, called MμV1, proposes a new local search method that disturbs a reduced number of the individuals' genes to avoid abruptly changing the direction of the algorithm. MμV1 also introduces a new operator for resetting the population when stagnation occurs. The new reset operator aims to conserve solutions that could have valuable information about the search space while helps the local search operator to reduce the amount of selected genes rapidly.
  - The second approach, called MμV2, introduces a new local search operator to perform a significant variation in the structure of the exploited neighbourhoods, provoking a major perturbation over the selected features (genes) of each solution. The objective is to speed up the discovery of new promising areas of the search space.
- Experiments are conducted using three state-of-the-art classifiers, for trying the proposals under different conditions.
- An in-depth analysis is carried out to demonstrate the capability of the MμGAs in comparison with other based metaheuristics techniques that were formerly used for the same problem instances.

The manuscript is organised as follows: Section 2 introduces the definition of the Feature Selection Problem and stochastic related works. Section 3 defines the memetic micro-Genetic Algorithm, the local search approaches, the novel reset population method, individual representation and operators of MμGA for feature selection. Experimental settings are described in Section 4, and the results are shown in 5. Finally, Section 6 summarises and remarks main results and conclusions.

2. Feature selection problem

There are several challenges in bioinformatics; one of them is to select groups of informative genes with high predictive power from existing samples. The most considerable difficulty in gene expression data analysis is the search space's high and disproportional dimension. Usually, gene expression data is presented in a matrix with a large number of genes and a small number of samples.

Feature selection is a combinatorial optimisation problem (NP-hard) (Xiong et al., 2001). Its objective is to eliminate characteristics that do not contribute to the classification problem or are redundant since they provide the same information. Feature selection for cancer data in microarray data consists of identifying relevant genes for classifying samples. A sample is generally catalogued as "no cancer" or "cancer" in a binary class dataset.

2.1. Problem definition

Let G be an $M \times N$ matrix with M samples and N features. The objective is to find a subset of n relevant features ($n \leq N$), so that classification processes may be improved.

Now, take x a binary vector with size N, where $x_i = 1$ denotes that feature i has been selected while $x_i = 0$ indicates that it has not been selected. The objective function to be maximised is shown in Eq. (1); it represents the fitness function that considers maximising the classification accuracy and reducing the number of selected characteristics when evaluating a solution x.

$$fitness(x) = \alpha \times accuracy(x) + (1.0 - \alpha) \times \left(1 - \frac{\sum_{i=1}^{N} x_i}{N}\right) \tag{1}$$

Where α is a weight with value 0.9, to control that the accuracy value takes precedence over the subset size. The *accuracy* is estimated running *k-Fold Cross-Validation* with a classifier (James et al., 2013). k-Fold Cross-Validation consists in dividing the data into k groups of similar size. $k - 1$ groups are used to train the model, and one group is employed to test — the process is repeated k times using a different group as the test set in each iteration. The procedure generates k estimations of error (test error); the average error test is used for the final estimation.

2.2. Literature review

Feature selection (FS) aims to identify the most relevant features from a dataset. Different areas of work use FS due to its ability to provide interpretability to the analysis of huge datasets (Sarker, 2021). For example, in bioinformatics, FS is commonly used to select the subset of genes that participate in a given biological process, from an immense number of genes (Bommert et al., 2022). There are three major approaches to FS: filter, wrapper and embedded methods (Agrawal et al., 2021)

Filter methods decide the subset of features to maintain based on statistical concepts like covariance of the features. One example is the paper developed by Abdi and Ghodsi (2020) which uses the distance correlation for deciding the most relevant subset of genes for microarray cancer data. Another example is presented in (Ke et al., 2018), where it is proposed a combination between the Symmetrical Uncertainty metric and the ReliefF algorithm. Onan and KorukoGlu (2015) introduce an approach which combines several filter methods by using genetic algorithms for enhancing text sentiment classification. Filter methods considered are information gain, relief algorithm, gain ratio, statistical probabilistic significance and statistical metrics like Chi-Square, Pearson correlation and symmetrical uncertainty coefficient. Results confirmed that the approach improves other individual filter methods. The weakness related to filter methods is that they generally just focus on specific relationships between variables, ignoring that features do not necessarily must have a relationship with other features to be relevant in the classification task.

Machine learning (ML) algorithms have called the attention of researchers due to their ability for pattern recognition in data. Tasks of ML like classification (Onan, 2019, Xue et al., 2022), regression (Austin et al., 2022), text analysis (Onan, 2018) and sentiment analysis (Onan, 2022), image classification (Xue & Qin, 2022) and cluster analysis (Li et al., 2021), highly contribute to understanding data and finding implicit patterns difficult to discover at a glance. Methods that perform FS during the training of the ML are known as Embedded Method (Rostami et al., 2021). Wrapped Methods are those FS approaches that use ML algorithms for assessing how relevant a subset of features yielded by another agent is.

In literature, a common wrapped method approach is to use metaheuristics for selecting features (leveraging its ability of exploring and exploiting the search space efficiently) and ML algorithms for validate the selected genes. There is a wide variety of metaheuristics that were used for feature selection. Some examples of these are the works in (Rao et al., 2019, Sayed et al., 2018, Anand & Arora, 2019), where a Canonical Artificial Bee Colony Algorithm, a Chaotic Dragonfly algorithm and a Chaotic Selfish Herd Algorithm were respectively proposed. These works use K-Nearest neighbours (K-NN), Support Vector Machines (SVM) and Decision Trees (DT) as methods that determine the relevance of the features. These methods have shown new alternatives to deal with the FS, reaching a good performance either in accuracy and retained features, compared with other methods. On the other hand, Boughaci and Alkhawaldeh (2018) proposed a study in that three Local Search Based methods were compared, namely, Hill Climbing method, Stochastic Method and Variable Neighbourhood Search method. Although these are methods based on local search, results have proved good performance in feature selection.

Genetic Algorithms (GA) are considered one of the first metaheuristics applied to FS. Nowadays, many works have been built based on GAs or have made modifications over them. Jadhav et al. (2018) proposed a Hybrid method with an Information Gain algorithm used to filter datasets and a GA as a part of a wrapper method, with SVM, K-NN and DT as classifiers. Ma and Xia (2017) introduced the use of a Tribe Competition-Based Genetic Algorithm for feature selection in Pattern Recognition. In (Vijayanand et al., 2018) a wrapped method using a GA is used to filter features for an intrusion detection dataset. Results show that the approach could improve the accuracy of a Support Vector Machine. Thus, it is possible to say that GA-based algorithms have demonstrated to provide great performance in selecting determinant features over datasets, which is likely associated with the robustness of the GA in such problems.

Although the above-mentioned algorithms are efficient in FS problems, they can have difficulties when making fine-tune adjustments, especially when their solutions are near to the optimal point of the search space. Memetic Algorithms (MA) point to solve this problem by integrating a global search component such as an evolutionary framework and a local search component. The first component enhances exploration, while the second one improves the exploitation of the search space. Several researchers have addressed FS by using MAs. In (Tubishat et al., 2020), a Dynamic Salp Swarm Algorithm has been improved by adding an initialisation process based on the Opposition Based Learning and a novel local search method. This work exhibited better exploration ability than other optimisation algorithms in accuracy and the number of selected features. Memetic variants of genetic algorithms and its derivatives have also taken place in several proposals, such as GA integrated with PSO (Moslehi & Haeri, 2019) and multi-objective CHC GA (Rathee & Ratnoo, 2020). Both of them were applied over feature selection benchmark datasets, and have shown promising results.

When FS must be performed over microarray cancer datasets, a new complication emerges. In this case, metaheuristics have to deal with many features and a few samples; the opposite of an ideal scenario. Several approaches were applied over microarray datasets. For example, Shukla (2019) proposed a hybrid feature selection method using a multi-population adaptive genetic algorithm and a filter method based on F-Score. Its results have shown that it can improve or at least equalise results of the state-of-the-art approaches. However, its fitness function only considers the accuracy provided by Support Vector Machine (SVM) and Naive Bayes classifiers, that suggests that the number of selected features did not influence the evolution. Alanni et al. (2019) also proposed a variation of the genetic algorithms that have reached good results either in accuracy and number of selected features, but they have evaluated their model using only an SVM with a unique configuration. In (Bilen et al., 2020) authors provided a Hybrid FS approach in that several filter methods were combined with genetic algorithms. Its experiments are applied over a leukaemia cancer dataset. Results have shown that the proposed method can provide a reduced number of selected features and a precision value near to 100%, but the model is tested only in one dataset.

Considering the advantages of MA, several authors have proposed to use them for FS over microarray data. In (Yang et al., 2008), a hybrid feature selection method is presented. The filter method is the Relief method, and the wrapper method is a memetic algorithm that combines a GA with a simple random local search algorithm. Results have shown that it can improve the performance of both Canonical GA and Canonical PSO. However, the model was tested by just one classifier, so it is uncorroborated in the efficiency over another classification configuration. On the other hand, the proposed local search method seems to do a little diversification and intensification between neighbours that appear not to minimise the number of selected features that achieve an acceptable accuracy value.

In (Rojas et al., 2020) a Cellular Genetic Algorithm (CGA) with a local search method specially designed is implemented, and three datasets are used for comparison. Pragadeesh et al. (2019) proposed a Hybrid FS method that uses a Canonical Micro Genetic Algorithm combined with an Information Gain filter method applied over cancer datasets. In this work, the authors did not propose a local search method to enhance search space exploitation, which probably had an incidence in the slight features reduction performed. Furthermore, the test was applied using SVM as a unique classifier.

In this paper, two novel Memetic Micro Genetic Algorithms for FS in a microarray data context are presented. They are combined with two specially designed local search methods. To the best of our knowledge, no presented work encompasses all the components included in our proposal.

3. Memetic micro-genetic algorithm (MμGA)

The field of evolutionary computation has emerged in the last decades to solve problems of different academic and industrial nature, due to these methods' ability to explore the space of possible solutions in-depth while trying to avoid exhaustive searches. However, in most cases, evolutionary algorithms suffer from low convergence rate depending on the problem. A memetic algorithm (Moscato et al., 1989, Moscato & Cotta, 2018) can be applied to improve this situation, incorporating a local search into the general evolutionary process. A synergistic behaviour is generated where each algorithm's weaknesses can be improved by combining with other techniques, allowing significant improvements without exhaustive search.

This section describes our memetic approach based on a micro-Genetic Algorithm (MμGA). The evolutionary process is performed by a micro-Genetic Algorithm (μGA) combined with a local search based on Hill Climbing algorithm (HC). Two variations of HC are presented for the FS problem. Section 3.1 describes the canonical μGA algorithm. Section 3.2 describes the local search approaches, and Section 3.3 reports the novel reset population. Section 3.4 presents the general scheme of our approach and specific details about the design.

3.1. Micro-genetic algorithm

Genetic algorithms (GAs) are a well-known and powerful tool used in optimisation problems and searching tasks. They are based on Darwin's theory of survival of the fittest (Holland, 1992). A variant of the GA known as Micro-Genetic Algorithm (μGA) provides some advantages over the conventional approach. The μGA is designed to work on minimal population size (p). This size is usually less than ten individuals (potential solutions of the searching problem) meanwhile, other evolutionary approaches work with 100 individuals (Krishnakumar, 1990).

A summary of the entire process of the μGA is described in Algorithm 1. As a first step, an initial population is randomly created and evaluated (in lines 2 and 3). Next, the evolutionary process applies the genetic operators (crossover and mutation) in line 6, and new solutions are evaluated in line 7. Then, the p individuals with the highest fitness value are selected, between the old and the new ones (line 8). The characteristics of the individuals tend to converge to the fittest solution over successive generations. To avoid this, the convergence of the population is evaluated. If all the solutions are more similar than a certain threshold, the best individual is saved, and the remaining are randomly created from scratch (line 10). This iterative process is repeated until the termination criterion is satisfied.

Using a small population enhances the computational resources usage, as a minimal number of solutions is in physical memory during the algorithm execution. This is especially important when working with high dimensional data (as microarray data sets) because memory overflow is prevented (Dokken & Fronk, 2018, Santiago et al., 2021). Saving computational resources also allows parallelism, which could significantly improve the process performance and reduce the execution times.

Another advantage of small populations is the high level of elitism during evolutionary process, due to the best solution of the population impacts heavily on the evolution of the other solutions. This provides the ability of a rapid convergence to optimal points (Jaen-Cuellar et al., 2016, Santiago et al., 2021). Additionally, by restarting the population constantly, not just diversity is achieved, but also the μGA has a better possibility of escaping from local optima, which results in better solutions (Madadi & Balaji, 2008).

The μGA algorithm has been used for diverse problems of academic and industry nature. These approaches have demonstrated their effectiveness in finding optimal (or near-optimal) solutions in landscapes with multiple local optima. Some problems can be found in the fields of oil (Ribas et al., 2013), construction (Au et al., 2003, Madadi & Balaji, 2008), aerospace (Chakravarty & Mittra, 2002, Truong et al., 2017), bioinformatics (Pragadeesh et al., 2019), energy (Burhan et al., 2016, Lee et al., 2019), sensor networks (Mendoza et al., 2007), urban planning (Chen & Song, 2012), airfoil (Szőllős et al., 2009), among others.

Algorithm 1 Pseudo-code of canonical μGA.

```
 1: procedure EVOLVE($popSize$)
 2:     $P \leftarrow GenerateInitialPopulation(popSize)$
 3:     $Evaluation(P)$                                  ▷ Evaluate initial population
 4:     while not StopCondition() do
 5:         $matingPop \leftarrow Selection(P)$
 6:         $offspringPop \leftarrow Reproduction(matingPop)$
 7:         $offspringPop \leftarrow Evaluation(offsrpingPop)$
 8:         $P \leftarrow selectBestSolutions(P, offspringPop)$
 9:         if $Convergence(P) < threshold$ then
10:             $P \leftarrow ResetPopulation(P)$
11:         end if
12:     end while
13:     return $bestSolution$
14: end procedure
```

3.2. Local search

The local search procedure is an individual reinforcement process to find a better solution around the best solutions (Blum et al., 2011). When an element is changed in a candidate solution, a neighbour chromosome (solution) is obtained.

Memetic algorithms introduce a local search (LS) operation to accelerate the whole evolutionary searching process. The local search algorithm starts from a candidate solution and then iteratively moves to a neighbour solution, making small perturbations. The process is repeated until a solution deemed optimal is found, or a specific number of movements l is completed. LS needs to be carefully designed to balance the acceleration of the convergence and the avoidance of the local optimum.

In this work, a Hill-Climbing algorithm (HC) based approach is used to increase the accuracy of the results of the μGA and to decrease the likelihood of getting stuck in suboptimal solutions. The relative simplicity of HC, its ability to improve solutions in short periods of execution time, and its adaptability to the problem make it a perfect component to be mixed with the μGA process. In this proposal, two HC-based local search designs are presented.

- The first variation of the HC algorithm is exposed in Algorithm 2. A random value between 0 and 1, called *prob*, is generated at the beginning of each iteration to choose a possible modification. There are four possible perturbations (each with an equal probability of being selected). The options allow modifying one, two or three variables located consecutively by $ConsecutivePerturb$ function, or altering five variables at random positions by $RandomPerturb$ function (Boughaci & Alkhawaldeh, 2018). These functions receive as parameters the solution to be perturbed and the number of variables to modify. The value of each variable can turn to false according to the probability presented in Eq. (2). When perturbations are applied, the resulting solution is conserved if it is better than the current. This LS alternative points not to disturb the solution so much and to ensure that individuals do not change abruptly in the search space. The aim is to improve the precision capacity by deactivating the selected genes and finding the most representative genes that were not considered during the evolutionary process. The memetic μGA that use this variation of HC is called MμV1.

$$1.0 - \frac{1}{N \times 0.2} \tag{2}$$

Algorithm 2 Pseudocode of the first variation of the HC.

```
1: function LocalSearch(Solution)
2:     $auxSolution \leftarrow Copy(Solution)$
3:     $nonContiguousChanges \leftarrow 5$
4:     $evals \leftarrow 0$
5:     while $evals < l$ do
6:         $prob \leftarrow Random(0, 1)$
7:         if $prob <= 0.25$ then
8:             $auxSolution \leftarrow ConsecutivePerturb(auxSolution, 1)$
9:         else if $prob > 0.25$ and $prob <= 0.5$ then
10:            $auxSolution \leftarrow ConsecutivePerturb(auxSolution, 2)$
11:        else if $prob > 0.5$ and $prob <= 0.75$ then
12:            $auxSolution \leftarrow ConsecutivePerturb(auxSolution, 3)$
13:        else
14:            $auxSolution \leftarrow RandomPerturb(auxSolution, NonContiguousChanges)$
15:        end if
16:        $Evaluate(auxSolution)$
17:        if $auxSolution.fitness > Solution.fitness$ then
18:            $Solution \leftarrow Copy(auxSolution)$
19:        end if
20:        $evals \leftarrow evals + 1$
21:    end while
22:    return $Solution$
23: end function
```

- The second variation can also randomly modify one, two or three variables arranged consecutively or alter 20% of their variables without requiring them to be contiguous. This occurs by changing the $nonContiguousChanges$ variable to the product between the number of genes N and 0.2 (in line 3 of Algorithm 2).
  This LS approach introduces the possibility of generating a stronger disturbance at some point in the evolutionary process. It seeks to explore a neighbourhood a little further from the current one, trying to find a significant improvement. However, the possibility of continuing to exploit nearby neighbourhoods remains. The memetic μGA hybridised with this local search variant is called MμV2.

3.3. Reset population

The first variation (MμV1) changes how the reset operator generates new solutions when the algorithm reaches nominal convergence. The objective is to contribute the local search operator by reducing even more the number of selected genes, considering that compact subsets of discriminative genes, related to diseases and disorders, are more valuable for biologist than having a big number of related genes (Yu, 2007).

The new reset operator takes the best solution to the new population, maintaining the elitism characteristics of the original reset operator. The difference is given when restarting the remaining solutions, which are modified by randomly deactivating 20% of their genes, according to probability presented in Eq. (2). An example of the resultant population after passing through the new reset operator is presented in Fig. 1

The advantage of the new MμV1 reset operator is that the solutions are not completely discarded, so that valuable information about the search process is not lost. Additionally, the new operator promotes finding a little set of genes that reaches high accuracy values by deactivating more genes than it activates. This modification can increase the exploration of the search space and accelerate the convergence to optimum points.

3.4. General scheme of MμGA approaches

This subsection describes the proposed MμGA. Fig. 2 shows the evolutionary process flow. The variations improve the exploitation ability of the μGA algorithm.

First, the algorithm creates and evaluates solutions. Then, the stop condition is check, if it is not fulfilled, the evolutionary process begins applying the genetic operators of μGA. Once the new population has been selected, a variation of the HC is applied. Next, the level of convergence of the population is evaluated. A few solutions can quickly converge to just one, trapping the population at a local optimum. A Hamming function is used to evaluate the nominal convergence of the solutions. If there is convergence, the population reset process is carried out. In the first variant, the change process explained in section 3.3 is applied. For the second variant, the canonical reset of μGA is used. In the memetic approach, the local search acts like an operator in μGA algorithm. The process is repeated until the stop condition is met, and the best solution is selected.

3.4.1. Chromosome representation

The encoding for FS in each chromosome is represented by a binary vector, where each position represents a feature in the dataset. Value *1* in the position i implies that the feature i is selected, whilst *0* indicates that i is not selected. Fig. 3 shows a representation of a vector solution with 8 features.

3.5. Time complexity analysis of the MμGA approaches

The analysis of the time complexity is carried out by using the big O notation. For evaluating the time complexity of the MμGA versions, several factors have to take into account: the population size (p), the number of genes of the solutions (N), the offspring size (V), the number of divisions for k-Fold Cross-Validation (k) and the number of evaluations performed in each execution of the local search operator (U). The total complexity of each version is obtained by considering the individual complexity of the following steps:

- The time complexity of the crossover operator, based on the complexity of the Half-Uniform Crossover, is $O(V \times N)$.
- The mutation operator complexity, based on the Bit-Flip Mutation, is $O(V \times N)$.
- The selection operator complexity is $O(p)$.
- The fitness evaluation has a time complexity of $O(k \times V \times N)$, where k is a constant ($k = 4$). So, $O(V \times N)$.
- The reset operator of the MμGA has a time complexity of $O(p \times N)$, given it, does not regenerate the best individual of the population.
- The local search operator has a time complexity of $O(p \times U \times N)$

The maximum complexity value from one of the steps gives the total algorithm complexity of both versions, namely MμV1 and MμV2. The

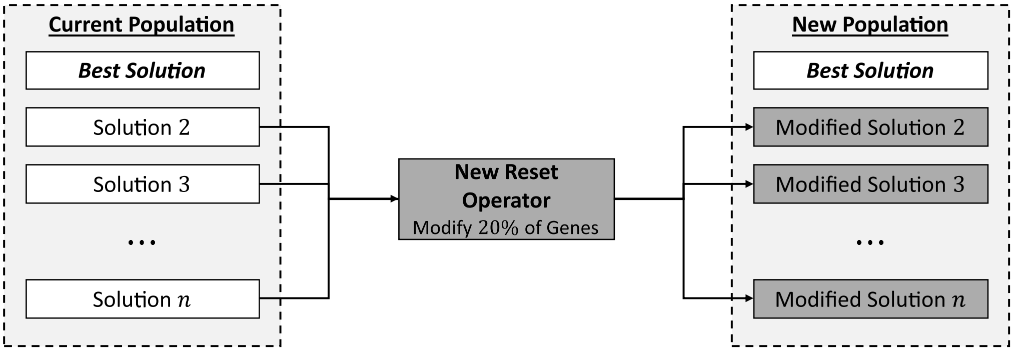


Fig. 1. New population after passing through the new reset operator.

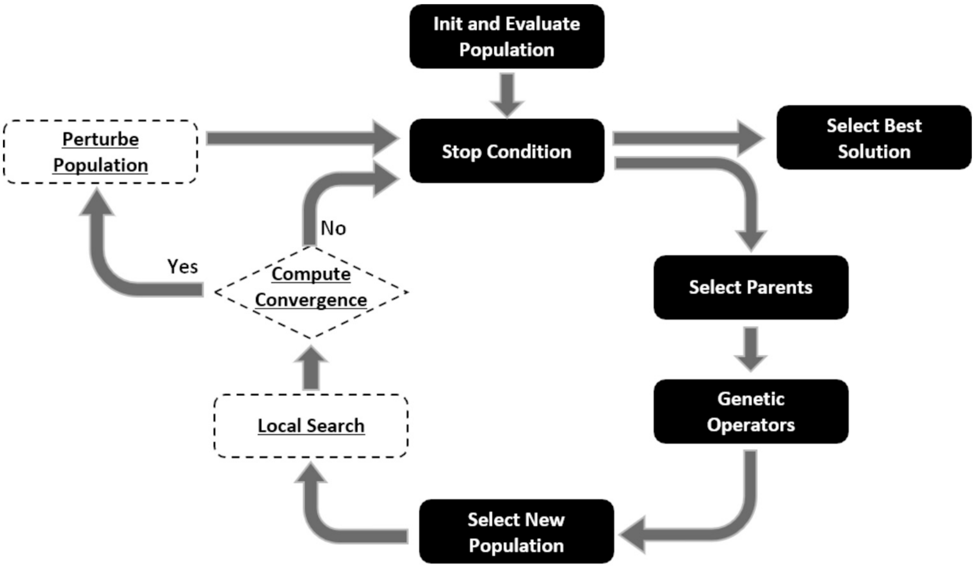


Fig. 2. Evolutionary process flow of MμGA.

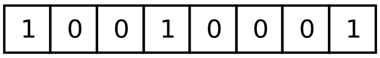


Fig. 3. Representation of chromosome encoding.

overall complexity is presented in Eq. (3), which also considerates the number of generations g (iterations) of the algorithm.

$$O(M\mu GA) = O(\max(g \times V \times N, g \times N \times V, g \times p, g \times V \times N, g \times p \times N, g \times p \times U \times N)) \tag{3}$$

This results in an adequate time complexity, given that the time complexities of solutions for NP-Hard problems are used to being in similar orders. Even more, NP-Hard problems do not require solutions to have in polynomial time complexities due to their inherent difficulty. Metaheuristics like MμGA versions are promising due to their ability to provide featured results independently of the instances being used (Oliveto et al., 2007).

4. Experimental settings

This section presents the experimental set-up. First, the datasets are introduced. Next, the methodology and the parameter settings in the tests are summarised.

4.1. Cancer datasets

Several experiments were performed with different instances of cancer datasets from the ELVIRA Biomedical Data Set Repository.[1] In this paper, seven datasets with samples belonging to two classes and one dataset with samples divided into four classes are used. Colon Cancer (Alon et al., 1999) distinguishes between normal samples and cancer. In Leukaemia (Golub et al., 1999), class 1 contemplates the samples classified as Acute Lymphoblastic Leukaemia (ALL) and class 2 encompasses Acute Myelogenous Leukaemia (AML) samples. Lung Cancer (Bhattacharjee et al., 2001) divides the samples into Adenocarcinoma (ADCA) and Malignant Pleural Mesothelioma (MLP). The Prostate dataset (Singh et al., 2002) separates data into two classes, tumour and normal instances. The Ovarian Cancer dataset (Petricoin et al., 2002) distinguishes between normal and ovarian cancer samples. Breast cancer (van't Veer et al., 2002) separates into Relapse samples after initial diagnosis and non-relapse ones. A summary of each case of study is shown in Table 1. Finally, the SRBCT dataset (Khan et al., 2001) is composed of 83 instances belonging to four different classes, Ewing Sarcoma (EWS), Burkitt Lymphoma (BL), Neuroblastoma (NB) and Rhabdomyosarcoma (RMS).

[1] https://leo.ugr.es/elvira/DBCRepository/.

Table 1
Cancer datasets summary.

Datasets	#Genes	#Samples	Classes

Colon Tumour	2000	62	Normal(22)
			Cancer(40)

Leukaemia	7129	72	ALL(47)
			AML(25)

Lung Cancer	12600	181	ADCA(150)
			MLP(31)

Prostate	12600	102	Tumour(52)
			Normal(50)

Ovarian Cancer	15154	253	Normal(91)
			Cancer(162)

Breast Cancer	24481	78	Relapse(34)
			Non-Relapse(44)

SRBCT	2308	83	EWS(29)
			BL(11)
			NB(18)
			RMS(25)

4.2. Classifiers

Three classifiers are explored to calculate the accuracy of each subset of features: Support Vector Machine, Decision Tree, and K-Nearest Neighbours. Scikit-learn is used to run them.[2]

Support Vector Machine (SVM) (Vapnik, 2000) is a supervised machine learning algorithm used for both classification and regression. It tries to identify the hyperplane placed in the middle between the classes, ensuring that the best separation is made. This makes SVM to focus on minimising the generalisation error instead of focusing on reducing the training error as other machine learning models. Nevertheless, there are situations in that no separation hyperplane exists between classes because of the dataset's distribution. Thus, it is necessary to use a mathematical trick called kernel that elevates the data to a dimension in which they can be separated (Noble, 2006). In this work, SVM is configured with a Radial Basis Function kernel (RBF); the parameter gamma is set in “scale” and the remaining parameters were defined taking the default configuration proposed by the library scikit-learn.

Decision Tree (DT) (Quinlan, 1986) is a supervised machine learning algorithm widely used because it gives interpretable results. A DT learns by building a hierarchical tree by partitioning the training set according to its attributes. The aim is to obtain pure subsets in the leaves of the tree; the subset only contains single class training data. Implicitly, a DT generates a set of rules that partition the dataset. In that way, a DT can provide an informative and robust hierarchical classification model. When the training phase finishes, test data passes through the nodes of the tree that decide what path the data has to follow until it reaches a leaf that determines its class (Wu et al., 2007, Myles et al., 2004). Default parameters proposed by the library scikit-learn were adopted for this algorithm.

K-Nearest Neighbours (K-NN) (Cover & Hart, 1967) is a supervised machine learning algorithm which determines the class of a test point, based in the predominance of a class in the k nearest training points. It is necessary to have a distance metric that allows estimating how far a test point is from the training points. It is also necessary to have a set of training point where each one is labelled, since they are going to be used to determine the class of a new test point. Finally, it is needed to define the value of the parameter k that is the number of the closest neighbours to be considered (Wu et al., 2007). In this work, k is set in 5, and the remaining parameters were configured with default values of the library scikit-learn.

4.3. Algorithms to compare

Several authors confirm the capability of different metaheuristics applied to the feature selection problem, and specifically over microarray datasets (Ali & Hassanien, 2015, Rojas et al., 2020). This section introduces the main characteristics of the techniques used for the analysis and comparisons in this work.

Canonical Genetic Algorithm (GA) (Holland, 1992) is a metaheuristic based on the Darwinian theory of evolution of species. The fundamental idea of GA is the survival of the fittest individual (a solution) of a population that evolves across time. First, the population is initialised and evaluated. The evaluation of each individual is performed by a fitness function that indicates how good is the solution regarding the problem. The population evolves by iterations of the GA called generations. In every generation, three evolutionary operators are applied to the population. The first one is the selection operator that selects a set of individuals to be kept in the next generation or well, to be recombined by the other operators. This operator is directly inspired by natural selection in the sense that fitter individuals are more likely to be selected. The crossover is the second operator applied to the population and consists of the exploitation of the shared space of two individuals selected by the selection operator. It combines the two individuals (called parents) to produce two new individuals (called children). Finally, the last operator is the mutation, that randomly changes genes of an individual to increase the diversity of the population. The mutation rate is usually set in a low value because a great number of mutations can provoke that the GA becomes in a primitive random search. The population evolves until the stop criterion is reached, then the best solution so far is returned as the best approximation for a given problem (Mirjalili, 2019).

Micro-Genetic Algorithm (μGA) (Krishnakumar, 1990) is a variation of the GA that emerges by the assumption that a few individuals are sufficient to reach an optimal solution for a given problem. μGA adopts all the concepts of the canonical GA, so it begins with a random population that evolves by generations where operators of selection, crossover and mutation are applied. The key differences are that it usually uses a small population such as 3, 5 or 7 individuals, it incorporates an operator to reset the population when the nominal convergence is reached (when all the individuals are very similar) and it has the aim to maintain the diversity of the population.

Binary Particle Swarm Optimisation (BPSO) (Kennedy & Eberhart, 1997) is an extension of the canonical Particle Swarm Optimisation algorithm (PSO) to allow it to deal with binary problems. PSO is based on the behaviour of different organisms, like bird flocking. It begins generating a swarm of particles, where each one is randomly positioned in different regions of the search space. At each iteration of the algorithm, the position and velocity of every particle are updated according to its previous position, and the position of the best-positioned particle until that moment. The key difference in BPSO is that the concept of velocity takes the form of the probability in that a bit or position becomes a one or a zero, and it is used when the position of the particle is updated (Lee et al., 2008). The Sigmoid function provides that probability.

Hill Climbing (HC) (Appleby, 1961) is considered the simplest approach to a local search method. Its strategy is to pass from a current solution to another better neighbour iteratively, to reach an optimal solution. It starts with a random solution; then, a perturbation is applied to identify a neighbour solution in each iteration. This neighbour is accepted only if it is better than the current solution (Al-Betar, 2016).

Variable Neighbourhood Search (VNS) (Hansen & Mladenović, 2018) is a variation of the HC algorithm based on the use of multiple neighbourhood structures rather than a single one. It changes systematically from one structure to another to quickly switch among different regions of the search space. First, it is necessary to define the structures to be used. Then, a random solution has to be generated. The following operation is repeatedly applied to this solution: shaking, that selects a neighbour solution based on a determined structure. Next, the hill-climbing operator discovers its neighbours without being restricted to

[2] https://scikit-learn.org/stable/.

Table 2
Algorithm configuration parameters.

Algorithm	Parameter	Value
BPSO	Population Size	30
	Position Updating	Sigmoid Function
GA	Population Size	100
	Crossover Operator	HUX - Prob.: 1.0
	Mutation Operator	Bit-Flip - Prob.: (1/N)
	Selection Operator	BTS
μGA	Population Size	5
	Crossover Operator	HUX - Prob.: 1.0
	Mutation Operator	Bit-Flip - Prob.: (1/N)
	Selection Operator	BTS
	Reset Population Threshold	5% Distinct Variables
HC	Changes per iterations	1 change
VNS	Neighbourhood Structures	1, 2 or 3 consecutive changes

any structure, in a given number of iterations. In every iteration, the current solution can be substituted if a new neighbour is better than itself. This process repeats until a stop condition is reached (Al-Betar, 2016).

4.4. Configuration of the algorithms

Experiments were done considering these five canonical approaches: Binary Particle Swarm Optimisation (BPSO), Genetic Algorithm (GA), Micro-Genetic Algorithm (μGA), Variable Neighbourhood Search (VNS) and Hill Climbing (HC). Table 2 shows the configuration for every implementation. For the μGA variations, the same configuration as for the canonical μGA was used.

In this work, the Binary Tournament Selection (BTS) (Deb, 2000) was used. This operator starts randomly selecting two parents from the population. Then, it selects the fittest one, which will be thrown to the next generation. The crossover operator is the Half Uniform Crossover (HUX) (Eshelman, 1991). Given two parents chosen by the selection operator, it swaps the half of the distinct genes between both of them. Finally, the mutation operator is the Bit-Flip Mutation. It works over each gene of the individual, mutating them (i.e. changing them from 0 to 1 and vice versa) with a given probability (Sivanandam & Deepa, 2008).

The threshold for the reset population operator of the μGA requires that at least 5% of the variables keep distinct, to avoid solutions converging to the same point in the search space. Both MμV1 and MμV2 perform 50 iterations per execution of the local search operator.

Due to the non-deterministic nature of the proposal, several independent runs (30) were considered for each algorithm and for each dataset. We have marked a result with a bold face when it is the best and with italic when it is the second best in performance. For the purpose of checking whether the differences between the algorithms are statistically significant or just a matter of chance, we applied the Wilcoxon rank-sum (Gibbons & Chakraborti, 2020) test and highlight, in the tables, the differences that are statistically significant. We always consider in this work a confidence level of 99% (i.e., significance level of $\alpha = 0.01$) in the statistical tests.

Experiments were performed in the Toko cluster (toko.uncu.edu.ar) with an AMD Opteron/Epyc processor (64 cores and 128 GB of RAM). The operating system is Ubuntu 18.04 LTS. Algorithms are implemented in jmetalpy (Benitez-Hidalgo et al., 2019).

5. Results

In this section, the analysis and results of the experiments are presented. First, in sections 5.1, 5.2 and 5.3, newly proposed approaches are compared to the most used methods in the literature for FS. The analysis considers fitness quality, execution time, accuracy, and the number of features. Finally, in section 5.4, a comparison with the results of several approaches applied to the same datasets is presented.

5.1. Classifiers and fitness analysis

Table 3 reports, for each algorithm and classifier, the average fitness value of the best solutions from the 30 independent runs. MμV2 overcomes all other algorithms (even MμV1) considering the three classifiers and all the datasets, excepting with the dataset SRBCT using the DT and K-NN classifiers, where MμV2 got the second-best value. This shows the capability of MμV2 to explore and exploit the search space, reaching highlighted solutions even under different distributions of classes of the datasets.

Moreover, comparing MμV2 and MμV1 confirms that the variation of a proportion of the number of genes made by MμV2 can improve the results. On its part, canonical μGA does not reach good results for any of the datasets confirming that the combination of reset and local search operators of the MμV1 and the local search operator of the MμV2 proposed in this work improve the exploitation and resolve the premature convergence of μGA.

Concerning the classifiers, the three of them achieve similar results of classification, but SVM yields the best fitness results. In the case of the Decision Tree, GA obtains the second-best results overcoming MμV1; whilst BPSO achieves the worst results for Leukaemia and Colon. VNS and HC fail to explore and exploit the search space, no matter the classifier.

Table 4 illustrates the details of the statistical analysis confirming the promising results of the MμV2. Values are presented in a table as an algorithm-by-algorithm comparison for each tested instance (one symbol for each dataset: Leukaemia, Breast, Colon, Lung, Ovarian, Prostate and SRBCT). A leftward triangle (◁) shows that the row setting gets statistically higher values than the column setting. In contrast, an upward triangle (△) indicates that the row setting gets statistically lower values than the column setting. If no significant differences are found, the place is completed with a dash (-). For example, the first upward triangle in the table corresponds to the significance of the difference between the performance metrics of BPSO against MμV1 using the SVM classifier in the Leukaemia case of study. It means that the difference of the performance in favour of the MμV1 (the orientation of the triangle) is significant (a triangle instead of a dash).

MμV2 is the only algorithm that obtains significant differences, winning in most of the performed tests, against all the other methods for the three Classifiers. MμV1 overcomes the other algorithms when implemented with SVM and K-NN classifiers. However, when Decision Tree is applied, MμV1 is overcome by the μGA and the GA. BPSO only defeats HC and VNS in two datasets, with SVM and K-NN Classifiers. All in all, we can confirm then that the MμV2 presents significant superior performance for all classifiers in all datasets, losing just against the GA when the DT is used to classify the SRBCT dataset.

Figs. 4a, 4b, and 4c show representative convergence curves of the algorithms for Ovarian instance. In order to understand the convergence at the beginning of the search process, the fitness value is plotted along the x-axis, which represents the percentage of fitness evaluations. As shown in the curves, in general, MμV2 has smoother curves and can reach a good convergence (high fitness value).

In Fig. 4b and 4c, MμV2 has a fast convergence at the beginning of the optimisation process and then starts to jump to higher fitness values. The other algorithms slow down in the latter process, unable to find significant improvements. The permeability jumps in the convergence curves of both memetic algorithms are owing to the application of the synergy between the two techniques used to find good solutions. After these jumps, the convergence slows down or even stops as the optimal solution is approached.

Table 3
Mean of the fitness quality indicator for SVM, DT and K-NN classifiers.

Class.	Alg.	Instance						
		Leukaemia	Breast	Colon	Lung	Ovarian	Prostate	SRBCT
SVM	BPSO	0.786	0.505	0.739	0.775	0.890	0.795	0.803
	GA	0.870	0.508	0.787	0.818	0.903	0.840	*0.851*
	μGA	0.789	0.502	0.764	0.781	0.890	0.804	0.826
	MμV1	*0.927*	*0.570*	*0.862*	*0.847*	*0.980*	*0.894*	0.846
	MμV2	**0.968**	**0.717**	**0.886**	**0.875**	**0.999**	**0.924**	**0.875**
	HC	0.769	0.512	0.839	0.763	0.903	0.735	0.848
	VNS	0.764	0.513	0.797	0.764	0.903	0.767	0.841
DT	BPSO	0.855	0.618	0.727	0.758	0.927	0.813	0.734
	GA	*0.881*	*0.653*	*0.785*	*0.795*	*0.930*	*0.854*	**0.838**
	μGA	0.880	0.641	0.767	0.787	0.929	0.851	0.822
	MμV1	0.873	0.622	0.741	0.777	0.928	0.840	0.798
	MμV2	**0.940**	**0.689**	**0.828**	**0.850**	**0.977**	**0.859**	*0.825*
	HC	0.867	0.608	0.734	0.752	0.926	0.806	0.773
	VNS	0.870	0.614	0.733	0.754	0.925	0.813	0.774
K-NN	BPSO	0.870	0.606	0.795	0.847	0.828	0.855	0.801
	GA	0.904	*0.650*	0.822	0.859	0.841	*0.880*	**0.877**
	μGA	0.876	0.608	0.805	0.847	0.828	0.855	0.812
	MμV1	*0.923*	0.624	*0.868*	*0.876*	*0.925*	0.851	0.859
	MμV2	**0.951**	**0.735**	**0.902**	**0.885**	**0.992**	**0.912**	*0.872*
	HC	0.831	0.574	0.813	0.844	0.839	0.793	0.800
	VNS	0.828	0.596	0.817	0.845	0.840	0.809	0.808

Table 4
Wilcoxon test results for the fitness differences (Leukaemia, Breast, Colon, Lung, Ovarian, Prostate and SRBCT) with significance level $\alpha = 0.01$.

Class.	Alg.	MμV1							MμV2						
SVM	BPSO	△	△	△	△	△	△	△	△	△	△	△	△	△	△
	GA	△	△	△	△	△	△	–	△	△	△	△	△	△	△
	μGA	△	△	△	△	△	△	△	△	△	△	△	△	△	△
	HC	△	△	–	△	△	△	–	△	△	△	△	△	△	△
	VNS	△	△	△	△	△	△	–	△	△	△	△	△	△	△
	MμV1								△	△	△	△	△	△	△
DT	BPSO	△	–	△	△	–	△	△	△	△	△	△	△	△	△
	GA	◁	◁	◁	◁	◁	◁	◁	△	△	△	△	△	△	◁
	μGA	◁	◁	◁	◁	–	◁	◁	△	△	△	△	△	△	–
	HC	–	–	–	△	△	△	△	△	△	△	△	△	△	△
	VNS	–	–	–	△	–	△	△	△	△	△	△	△	△	△
	MμV1								△	△	△	△	△	△	△
K-NN	BPSO	△	△	△	△	△	–	△	△	△	△	△	△	△	△
	GA	△	◁	△	△	△	◁	◁	△	△	△	△	△	△	–
	μGA	△	△	△	△	△	–	△	△	△	△	△	△	△	△
	HC	△	△	△	△	△	△	△	△	△	△	△	△	△	△
	VNS	△	△	△	△	△	△	△	△	△	△	△	△	△	△
	MμV1								△	△	△	△	△	△	△

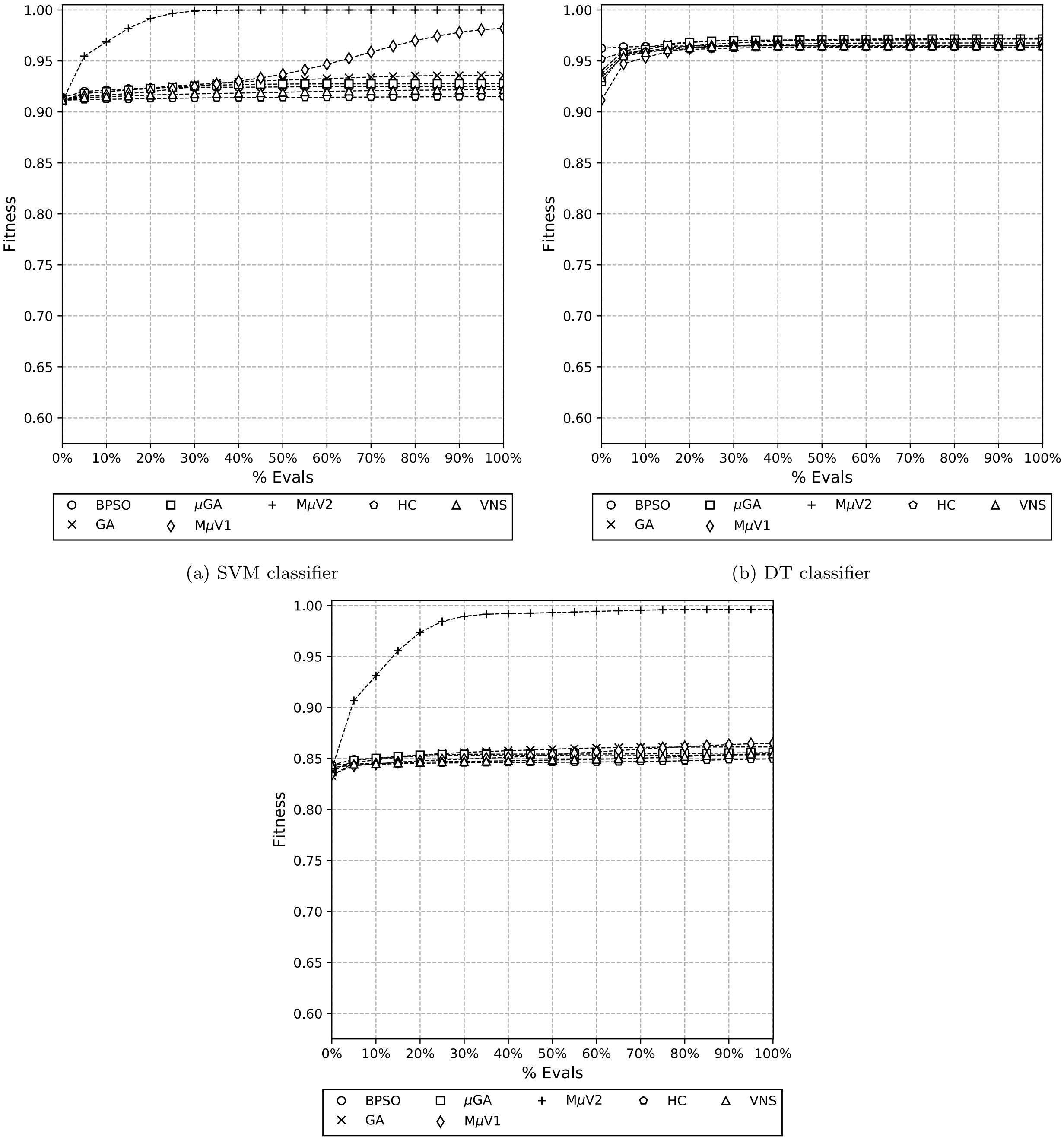
5.2. Time analysis

In order to analyse the complete performance of the approaches under the study, the results of the execution times are reported and discussed. Table 5 displays the average time consumed in seconds in each instance for the case of SVM, Decision Tree and K-NN. Column two present the name of each algorithm, and columns three to nine present the average time measured in seconds for each algorithm. MμV2 has obtained shorter times for most instances. In general, Table 5 indicates that the MμV2 approach obtains lower times with all the classifiers for all the instances. These times might be due to the intrinsic characteristics of certain operations in the MμGA, which have a high degree of parallelization that can maximise the efficiency of each thread and thus, the simplicity of each kernel is maintained.

It is observed that the times of the two versions MμGA are shorter than the rest of the algorithms. This behaviour is observed for both small and large instances, and even with multi-class datasets like SRBCT. Similarly, it is observed that the times of the two versions MμGA do not increase exponentially, but rather that, the times are reduced compared to the rest of the algorithms as the characteristics increase in each instance.

5.3. Classification accuracy and number of selected features

Table 6 shows the maximum and average values of the classification accuracy obtained by each algorithm and classifier for each dataset. The reported values correspond to the accuracy calculated for the best individual of the last run in each execution. MμV2 achieves the best results for the three classifiers. Moreover, MμV2 is the only one that obtains 100% of accuracy for Ovarian dataset in the three classifiers. It is interesting that for Ovarian cancer instance DT classifier obtains similar results in all algorithms. Similarly occurs for the rest of datasets, this may be because of the classifier and not because of a problem of convergence in the algorithms. The algorithms attain better accuracy with SVM and K-NN classifiers. In particular, when SVM is used, MμV1 earns second-best values. For K-NN, MμV2 achieves the best results for Colon and Lung instances maximum accuracy and the second place in Colon and Ovarian datasets average accuracy. Regarding the prostate dataset,



(a) SVM classifier

(b) DT classifier

(c) K-NN classifier

Fig. 4. Evolution of the fitness quality indicator for each algorithm using different classifiers for the Ovarian dataset.

the MμV1 and the MμV2 have overcome the other algorithms with all the considered classifiers. With the SRBCT dataset, the MμV2 has been positioned as the second-best value in both DT and K-NN classifiers, but it was the best result with the SVM classifier. Clearly, the special local search of MμGAs helps to exploit the solution space, and it is reflected in the accuracy.

Table 7 exhibits how MμV2 selects a few features from the total number of characteristics for each dataset, no matter the classifier. In particular, for Leukaemia, using SVM classifier (Table 7) chooses three features in the best execution and an average of five features. Indeed, this not affect the accuracy (Table 6) considering MμV1 reaches the best accuracy values for Leukaemia. Similarly occurs for the rest of the datasets except for Colon where MμV1 attains the minimum number of features selected with SVM classifier.

MμV2 takes the minimum number of features with DT and K-NN classifiers, followed by MμV1. Both MμV2 and MμV1 select far fewer features than the rest of the algorithms. This means that with few characteristics, both memetic versions can achieve good accuracy. For Ovarian cancer dataset, using K-NN classifier, the accuracy obtained by the memetic algorithms is similar but MμV1 needs several more features (187.9 avg.) to achieve this accuracy in comparison with MμV2 (5.8 avg.).

Figs. 5a, 5b, and 5c display, for Ovarian dataset, the evolution of the accuracy and the number of features selected during one execution of each algorithm with each classifier. As it can be seen, using the SVM Classifier (Fig. 5a), BPSO, GA and μGA require numerous features to gain accuracy. Clearly, these algorithms fail to get a good selection of relevant features. VNS and HC reach similar accuracy

Table 5
Mean of time quality indicator (in seconds) for SVM, DT and K-NN classifiers.

Classifier	Algorithm	Instance						
		Leukaemia	Breast	Colon	Lung	Ovarian	Prostate	SRBCT
SVM	BPSO	1950	9200	603	15300	15400	3637	670
	GA	1210	6500	381	14300	14200	2725	454
	μGA	1320	7500	404	14900	15300	2807	501
	MμV1	*787*	*3230*	283	*6550*	*6360*	*1457*	309
	MμV2	**653**	**2470**	*263*	**3630**	**3480**	**1131**	**264**
	HC	876	6120	**259**	10600	11300	2072	*282*
	VNS	903	5620	274	10600	11100	2033	312
DT	BPSO	2150	8940	815	10400	20600	3392	695
	GA	1550	7300	647	10000	20400	2641	546
	μGA	1650	7500	654	10400	21100	2653	564
	MμV1	*1070*	*4920*	510	*5240*	*9260*	*2378*	*510*
	MμV2	**909**	**2750**	*490*	**2860**	**5030**	**1678**	**415**
	HC	1170	6210	**489**	7470	16800	2397	521
	VNS	1230	6220	513	7400	16600	2435	*510*
K-NN	BPSO	2310	10300	740	*65200*	11000	4218	976
	GA	1450	7730	493	70700	9460	3347	790
	μGA	1480	7650	494	71000	9520	3493	812
	MμV1	1260	6700	*432*	69900	*9320*	*2370*	620
	MμV2	**814**	**3650**	**314**	**13300**	**3140**	**1608**	**570**
	HC	*1240*	6930	441	72000	9460	2650	*605*
	VNS	1260	*6660*	437	70500	9380	2632	644

Table 6
Maximum and average values of accuracy for algorithms and instances with the SVM, DT and K-NN classifiers.

Class.	Alg.	Colon		Leukaemia		Lung		Ovarian		Breast		Prostate		SRBCT	
		Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg
SVM	BPSO	0.77	0.76	0.84	0.81	0.78	0.77	0.93	0.93	0.50	0.50	0.81	0.80	0.83	0.80
	GA	0.81	0.80	*0.94*	0.90	0.82	0.81	0.94	0.94	0.50	0.50	*0.86*	0.84	*0.86*	*0.85*
	μGA	0.80	0.77	0.84	0.81	0.78	0.78	0.93	0.93	0.50	0.50	0.82	0.81	0.84	0.83
	MμV1	*0.88*	*0.85*	**0.98**	*0.92*	*0.88*	*0.85*	*0.99*	*0.98*	*0.57*	*0.52*	**0.94**	*0.89*	*0.86*	*0.85*
	MμV2	**0.89**	**0.87**	**0.98**	**0.96**	**0.89**	**0.87**	**1.00**	**1.00**	**0.77**	**0.69**	**0.94**	**0.92**	**0.90**	**0.87**
	HC	*0.88*	0.82	0.81	0.76	0.79	0.76	0.92	0.92	0.50	0.59	0.79	0.73	*0.86*	*0.85*
	VNS	0.81	0.78	0.82	0.76	0.79	0.76	0.93	0.92	0.49	0.49	0.81	0.77	*0.86*	0.84
DT	BPSO	0.74	0.67	0.86	0.81	0.76	0.75	*0.97*	*0.96*	0.62	0.57	0.84	0.82	0.79	0.74
	GA	*0.77*	*0.72*	0.90	*0.85*	*0.80*	*0.79*	*0.97*	*0.96*	**0.66**	**0.60**	**0.87**	**0.86**	**0.88**	**0.84**
	μGA	0.74	0.68	0.90	0.83	*0.80*	0.78	*0.97*	0.95	*0.63*	*0.59*	**0.87**	*0.85*	0.85	*0.83*
	MμV1	0.72	0.67	*0.92*	0.84	0.79	0.77	*0.97*	*0.96*	0.61	0.57	**0.87**	0.84	0.85	0.80
	MμV2	**0.84**	**0.75**	**0.93**	**0.92**	**0.88**	**0.85**	**0.98**	**0.97**	**0.66**	**0.60**	**0.87**	**0.86**	*0.86*	*0.83*
	HC	0.69	0.64	0.87	0.80	0.77	0.75	*0.97*	0.95	0.60	0.56	*0.86*	0.81	0.85	0.78
	VNS	0.73	0.64	0.85	0.76	0.77	0.75	*0.97*	0.95	0.60	0.56	*0.86*	0.82	0.85	0.78
K-NN	BPSO	0.83	0.82	0.92	0.91	0.85	0.84	0.86	0.86	0.63	0.61	0.87	0.86	0.82	0.80
	GA	0.85	0.84	*0.96*	*0.94*	*0.86*	*0.85*	0.87	0.87	*0.68*	*0.66*	0.89	*0.88*	**0.90**	**0.88**
	μGA	0.84	0.82	0.92	0.91	0.85	0.84	0.86	0.86	0.63	0.62	0.88	0.86	0.85	0.81
	MμV1	**0.90**	*0.86*	0.95	0.92	**0.89**	**0.88**	*0.94*	*0.92*	0.63	0.60	*0.91*	0.85	**0.90**	0.86
	MμV2	**0.90**	**0.89**	**0.97**	**0.95**	**0.89**	**0.88**	**1.00**	**0.99**	**0.80**	**0.71**	**0.94**	**0.91**	**0.90**	*0.87*
	HC	0.85	0.80	0.89	0.83	0.85	0.84	0.86	0.85	0.59	0.57	0.83	0.79	*0.86*	0.80
	VNS	*0.87*	0.81	0.88	0.83	0.85	0.84	0.86	0.85	0.61	0.59	0.84	0.81	*0.86*	0.81

than the previous algorithms with relative fewer features. Only MμV1 and MμV2 achieve a balance between numbers of features selected and high accuracy. In the case of Decision Tree classifier (Fig. 5b), MμV2 attains good accuracy with fewer features in comparison with the other algorithms. If we compare MμV2 and MμV1, evidently the fact that changes in the solutions are less restricted than MμV1 ensures a balance between exploration and exploitation, reflected in the accuracy reached and on the features that were selected. For K-NN classifier (Fig. 5c), the number of selected features for BPSO, GA and μGA are similar and all of them obtain low accuracy in comparison with HC and VNS, which gain similar accuracy with fewer features. Once again, MμV1 and MμV2 overcome the other algorithms. However, only MμV2 obtains, with few selected features, 100% of accuracy.

5.4. Comparison with state-of-the-art

This section compares the best accuracy and number of selected features arrived by the MμGA versions against state-of-the-art proposals using different classifiers. Results are distributed in three tables, each corresponding to a different classifier.

The algorithms from the state-of-the-art considered for comparisons are: the Binary Differential Evolution (BDE) (Apolloni et al., 2016), the Penguin Search Optimisation Algorithm with Rapid Convergence (PeSOA-C) (Dif et al., 2018), the hybridisation between Information Gain and the Micro-Genetic Algorithm (IG-μGA) (Pragadeesh et al., 2019), the Multi-Objective Spotted Hyena Optimiser, hybridised with the Salp Swarm Optimisation (C-HMOSHSSA) (Sharma & Rani, 2019) and the Ant Colony Optimisation - Selection (ACO-S) (Li et al., 2013). A summary of their configurations is presented in Table 8. It informs

Table 7
Min and average value of number of selected features for each algorithm and instance with the SVM, DT and K-NN classifiers.

Class.	Alg.	Colon		Leukaemia		Lung		Ovarian		Breast		Prostate		SRBCT	
		Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg
SVM	BPSO	819	864.5	3183	3271.9	5674	5834.6	6479	6788.6	9977	10430.8	5316	5628.6	941	1011.4
	GA	552	591.0	2602	2756.1	5280	5469.8	5954	6239.4	9061	9314.2	5135	5364.5	622	677.1
	μGA	596	660.5	2886	3069.3	5566	5730.4	6680	6800.6	10821	10972.0	5621	5734.8	722	797.9
	MμV1	**3**	*16.1*	*27*	*113.4*	*88*	*273.1*	*52*	*128.9*	*28*	*111.5*	*56*	*172.2*	*21*	*67.0*
	MμV2	*4*	**10.2**	**3**	**5.2**	**6**	**37.1**	**3**	**6.5**	**3**	**8.2**	**4**	**13.4**	**6**	**21.7**
	HC	27	51.5	853	910.5	2802	2880.6	3841	3951.4	8045	8163.8	2756	3074.8	42	71.2
	VNS	146	190.8	1311	1347.2	3071	3176.4	4109	4182.9	7942	8059.8	3119	3266.4	213	232.5
DT	BPSO	840	*879.9*	*3005*	*3118.6*	5403	*5571.9*	*6541*	*6697.8*	*10765*	*10900.8*	5439	*5561.3*	975	*1019.5*
	GA	930	960.5	3383	3441.2	6204	6265.6	7338	7475.9	11990	12135.5	6124	6265.2	1096	1144.8
	μGA	932	979.2	3360	3483.2	6134	6261.3	7384	7486.0	12101	12214.2	6119	6254.4	1066	1153.8
	MμV1	*770*	961.0	3197	3501.8	*5059*	5807.5	7481	7557.9	12167	12254.8	*4195*	6047.5	*946*	1135.6
	MμV2	**7**	**162.2**	**1**	**16.4**	**181**	**1069.8**	**2**	**175.7**	**611**	**2472.5**	**1052**	**2915.1**	**277**	**668.9**
	HC	936	982.3	3485	3537.5	6208	6305.1	7514	7610.2	12097	12219.3	6181	6281.8	1084	1153.8
	VNS	952	989	3456	3564.0	6210	6285.5	7496	7600.5	12128	12258.3	6152	6264.6	1110	1155.9
K-NN	BPSO	822	861.8	3115	3218.2	5459	5683.3	6716	6852.6	10726	11122.8	5385	5665.9	976	1032.8
	GA	493	603.5	2638	2814.3	5109	5391.5	6174	6467.9	11274	11534.6	5158	5455.5	826	904.8
	μGA	615	652	3006	3091.9	5662	5758.4	6752	6917.8	11550	11741.2	5611	5794.4	791	855.3
	MμV1	*11*	*37.8*	*48*	*252.4*	*356*	*890.5*	*57*	*187.9*	*1452*	*4212.1*	*303*	*1590.1*	*72*	*162.3*
	MμV2	**3**	**7.8**	**6**	**29.4**	**26**	**206.9**	**3**	**5.8**	**10**	**168.3**	**4**	**52.2**	**12**	**44.4**
	HC	25	105.1	929	1014.6	2895	3127.8	3967	4086.8	8430	8706.0	2859	3077.4	147	215.5
	VNS	155	202.8	1324	1415.6	3215	3352.2	4178	4321.1	8400	8765.1	3160	3331.2	310	401.0

Table 8
Configuration for state-of-the-art algorithms.

Algorithm	Datasets	Classifiers	Evaluations Performed
BDE	Colon, Leukaemia, Lung, Ovarian and Prostate	SVM, DT and K-NN	40000
PeSOA-C	Colon, Leukaemia, Lung, Ovarian, Breast and SRBCT	SVM	2500
IG-μGA	Lung, Ovarian	SVM	10000
C-HMOSHSSA	Leukaemia, Breast, Prostate and SRBCT	SVM, DT and K-NN	30000
ACO-S	Leukaemia and Colon	DT and K-NN	8000

Table 9
Comparison with SVM wrapper methods. Accuracy and number of selected features of the best found solution (between brackets).

Algorithm	Instance						
	Colon	Leukaemia	Lung	Ovarian	Breast	Prostate	SRBCT
BDE	0.75(3)	0.88(8)	*0.99(3)*	*0.99(2)*	-	*0.94(3)*	-
PeSOA-C	**0.92(127)**	**1.00(301)**	0.98(613)	**1.00(355)**	*0.81(2036)*	-	**1.00(52)**
IG-μGA	-	-	**1.00(1706)**	0.95(2844)	-	-	-
C-HMOSHSSA	-	**1.00(4)**	-	-	**1.00(3)**	**0.97(4)**	**1.00(4)**
MμV1	0.88(3)	*0.98(27)*	0.88(88)	*0.99(52)*	0.57(28)	*0.94(56)*	0.86(21)
MμV2	*0.89(4)*	*0.98(3)*	0.89(6)	**1.00(3)**	0.77(3)	*0.94(4)*	*0.90(6)*

the datasets and classifiers used and the number of fitness evaluations performed.

Table 9 compares the results of the MμGA versions against the BDE, the PeSOA-C, the IG-μGA and the C-HMOSHSSA, using the SVM classifier. Both MμV1 and MμV2 obtained comparable results with the state-of-the-art approaches. MμV2 has shown it can reach high values of accuracy, utilising a minimum number of selected features. Although the PeSOA-C has shown high accuracy values in most of the datasets evaluated, it performs a minor reduction of the selected features, which could be related to the number of evaluations performed. Keeping a significant number of selected features could lead to increased accuracy, which can explain that behaviour.

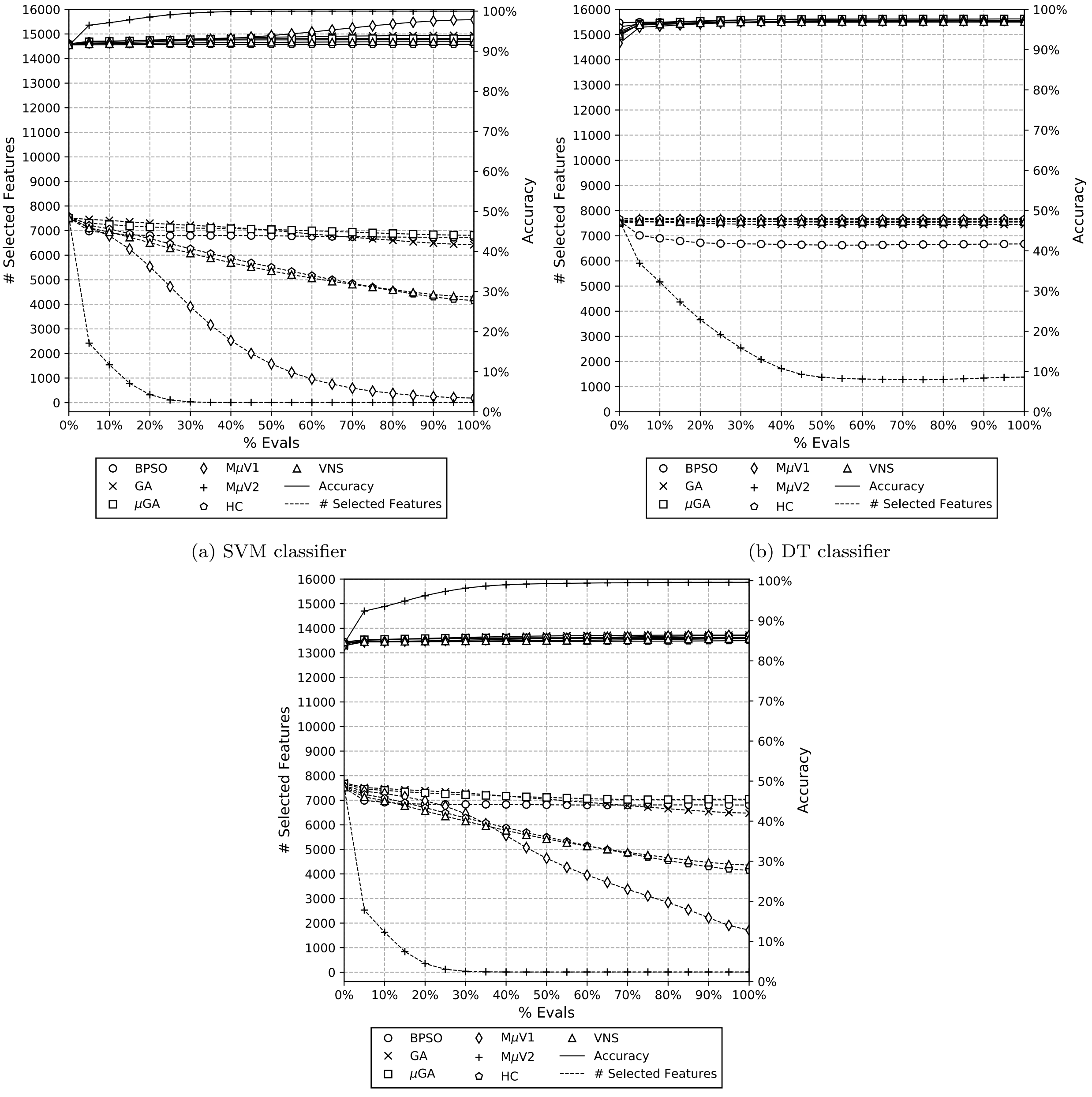
While the BDE produces results very similar to those of the MμGA, it involves a lot more evaluations, suggesting that it needs a major time and resource consumption to achieve such accuracy values and reduction of dimension. The IG-μGA, which makes the same number of evaluations as the MμGA versions, provided a large number of selected genes.

Compared to the state-of-the-art algorithm, the MμGA versions shown a competitive behaviour, founding the minimal number of selected features that provides a high accuracy.

Table 10 compares approaches that use DT as the classifier. Considered algorithms are the BDE, the ACO-S and the C-HMOSHSSA. It appears that the performance of the MμGA has been reduced, showing an increase in the number of selected features and a decrease in the value of accuracy. Nevertheless, most of the accuracy values stayed near to the state-of-the-art approaches, exhibiting similar performance with every dataset. Moreover, MμV2 has stood out, achieving the best results in three out of the seven datasets, and reaching the second-best result in three out of the remaining. This reflects that the MμV2 can be a competitive variant when the DT classifier is used.

The increment in the number of selected features could indicate that the MμGA versions find it easier to work with classifiers which base their classifications on distances of the points in the feature space, instead of those that rely on the tree-like model of decisions.

Finally, in Table 11 algorithms that use K-NN classifier are compared. Again, BDE, ACO-S and C-HMOSHSSA are used for comparisons.



(a) SVM classifier

(b) DT classifier

(c) K-NN classifier

Fig. 5. Evolution of the accuracy and the number of selected features for each algorithm with different classifier for the Ovarian dataset.

Table 10
Comparison with DT wrapper methods. Accuracy and number of selected features of the best found solution (between brackets).

Algorithm	Instance						
	Colon	Leukaemia	Lung	Ovarian	Breast	Prostate	SRBCT
BDE	**0.88(2)**	**0.94(1)**	**0.97(1)**	**0.98(1)**	-	*0.74(2)*	-
ACO-S	0.82(75)	0.91(93)	0.87(141)	-	-	-	-
C-HMOSHSSA	-	**0.94(4)**	-	-	**0.96(4)**	0.32(4)	0.80(4)
MμV1	0.72(770)	0.92(3197)	0.79(5059)	0.97(7481)	0.61(12167)	**0.87(4195)**	*0.85(946)*
MμV2	*0.84(7)*	*0.93(1)*	*0.88(181)*	**0.98(2)**	*0.66(611)*	**0.87(1052)**	**0.86(277)**

In this case, results have shown that MμV2 reached good results in most of the instances, achieving the second-best position in four out of the seven datasets and the best solution in two opportunities, suggesting a good performance. Likewise, an acceptable balance between accuracy and the number of selected features was obtained.

The MμV2 was slightly outperformed or equalled in Colon, Leukemia, Lung and Ovarian datasets by the BDE or C-HMOSHSSA, which performed more fitness function evaluations during their executions. It appears, based on evidence, that MμV2 was more efficient at arriving at optimal solutions than other approaches.

Table 11
Comparison with K-NN wrapper methods. Accuracy and number of selected features of the best found solution (between brackets).

Algorithm	Instance						
	Colon	Leukaemia	Lung	Ovarian	Breast	Prostate	SRBCT
BDE	*0.88(3)*	*0.97(2)*	**0.99(2)**	*0.99(2)*	-	**0.97(3)**	-
ACO-S	0.80(91)	0.89(101)	0.88(138)	-	-	-	-
C-HMOSHSSA	-	**1.00(4)**	-	-	**0.92(4)**	**0.97(4)**	**1.00(4)**
MμV1	**0.90(11)**	0.95(48)	*0.89(356)*	0.94(57)	0.63(1452)	0.91(303)	*0.90(72)*
MμV2	**0.90(3)**	*0.97(6)*	*0.89(26)*	**1.00(3)**	*0.80(10)*	*0.94(4)*	*0.90(12)*

Considering all the analysis, it is possible to conclude that the two memetic proposals (MμV1 and MμV2) reach comparable results with other start-of-the-art approaches and demonstrate that they can obtain high accuracy values whilst keeping a reduced number of selected features. It is essential to highlight that these results are observed independently of the classifier being used, which ensures that the algorithm's procedure can find the more relevant features under different conditions of classification.

6. Conclusions

Memetic models are an appropriate approach for integrating different techniques that explore and exploit the search space. The present work proposes two memetic algorithms (MμV1 and MμV2) that fusion a micro-genetic algorithm and a hill-climbing local search procedure for the feature selection problem. The work's main contributions are the hybrid model, which use two novel local search strategies and a new reset population procedure with a specific focus on solving cancer microarray data feature selection. Both proposals comprise making intuitive exploitation of the search space by achieving a balance between the application of small and strong variations of the active features. Three classifiers (Support Vector Machine, Decision Tree and K-Nearest Neighbours) were used as wrapper methods. The performance of the proposed approaches was applied over seven cancer datasets.

To analyse the potential of the algorithms, exhaustive analysis and comparison with other techniques were made. The contrast was carried out with popular literature algorithms: Genetic Algorithm, canonical Micro-Genetic Algorithm, Binary Particle Swarm Optimisation, Hill Climbing and Variable Neighbourhood Search. Later, a comparison was made with state-of-the-art approaches that consider the same datasets as the present work. In this context, with all the classifiers, MμV2 obtains excellent results of accuracy values and the number of selected features.

In terms of fitness values, MμV1 and MμV2 overcome the rest of the canonical algorithms for all datasets, whatever the classifiers were used. This tendency was statistically confirmed. In particular, the results clearly show the efficiency of MμV2. Also, the accuracy and number of selected features analysis were done. The accuracy reached by MμV2 overcomes the rest of the algorithms without a detriment of the number of selected features. Indeed, MμV2 needs fewer features to reach its excellent results.

In terms of the execution time, a great reduction in time is observed (a gain of time up to 4 times approximately) concerning the longest time obtained for the rest of the algorithms. In the same way, compute time scales linearly when solving high-dimension problem instances.

In general, the memetic approaches show excellent performance on the instances tested, providing better computational results than well-known algorithms to resolve the problems. Moreover, the results show that the second approach (MμV2) explores and exploits the search space efficiently, finding excellent quality solutions at reasonable execution times for the types of problems treated.

For future work, the behaviour of the approach to parallel or distributed computing platforms will be evaluated. It is also desirable to perform a sensitivity analysis to observe whether the best performance of the MμV2 is reached by deactivating 20% of the genes or using a different proportion of genes. Regarding MμV1, future works point to find a configuration of the reset operator to effectively contributes the behaviour of the local search. Finally, it would be interesting it seeks to expand these new local searches capacity of to other types of combinatorial problems.

CRediT authorship contribution statement

Matías Gabriel Rojas: Conceptualization, Formal analysis, Investigation, Methodology, Software, Writing – original draft, Writing – review & editing. **Ana Carolina Olivera:** Conceptualization, Formal analysis, Investigation, Methodology, Validation, Writing – original draft, Writing – review & editing. **Jessica Andrea Carballido:** Conceptualization, Formal analysis, Investigation, Methodology, Validation, Writing – original draft, Writing – review & editing. **Pablo Javier Vidal:** Conceptualization, Formal analysis, Investigation, Methodology, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by Universidad Nacional de Cuyo (Project No. 06/B052-T1) and Universidad Nacional del Sur, Secretaría General de Ciencia y Tecnología (grant PGI No. 24/N052). Authors thank Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina.

References

Abdi, L., & Ghodsi, A. (2020). Discriminant component analysis via distance correlation maximization. *Pattern Recognition*, *98*.

Abu Zaher, A., Berretta, R., Noman, N., & Moscato, P. (2019). An adaptive memetic algorithm for feature selection using proximity graphs. *Computational Intelligence*, *35*(1), 156–183.

Agrawal, P., Abutarboush, H. F., Ganesh, T., & Mohamed, A. W. (2021). Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019). *IEEE Access*, *9*, 26766–26791.

Al-Betar, M. A. (2016). β-hill climbing: An exploratory local search. *Neural Computing & Applications*, *28*(S1), 153–168.

Alanni, R., Hou, J., Azzawi, H., & Xiang, Y. (2019). A novel gene selection algorithm for cancer classification using microarray datasets. *BMC Medical Genomics*, *12*(10).

Ali, A. F., & Hassanien, A.-E. (2015). A survey of metaheuristics methods for bioinformatics applications. In *Intelligent systems reference library* (pp. 23–46). Springer International Publishing.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., & Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, *96*(12), 6745–6750.

Alonso-Betanzos, A., Bolón-Canedo, V., Morán-Fernández, L., & Seijo-Pardo, B. (2019). Feature selection applied to microarray data. In *Methods in molecular biology* (pp. 123–152). New York: Springer.

Anand, P., & Arora, S. (2019). A novel chaotic selfish herd optimizer for global optimization and feature selection. *Artificial Intelligence Review*, *53*(2), 1441–1486.

Apolloni, J., Leguizamón, G., & Alba, E. (2016). Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments. *Applied Soft Computing*, *38*, 922–932.

Appleby, J. S. (1961). Techniques for producing school timetables on a computer and their application to other scheduling problems. *The Computer Journal*, *3*(4), 237–245.

Au, F., Cheng, Y., Tham, L., & Bai, Z. (2003). Structural damage detection based on a micro-genetic algorithm using incomplete and noisy modal test data. *Journal of Sound and Vibration*, *259*(5), 1081–1094.

Austin, D. E., Lee, D. S., Wang, C. X., Ma, S., Wang, X., Porter, J., & Wang, B. (2022). Comparison of machine learning and the regression-based EHMRG model for predicting early mortality in acute heart failure. *International Journal of Cardiology*.

Benitez-Hidalgo, A., Nebro, A. J., Garcia-Nieto, J., Oregi, I., & Del Ser, J. (2019). jMetalPy: A python framework for multi-objective optimization with metaheuristics. arXiv preprint, arXiv:1903.02915.

Bhattacharjee, A., Richards, W. G., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., Loda, M., Weber, G., Mark, E. J., Lander, E. S., Wong, W., Johnson, B. E., Golub, T. R., Sugarbaker, D. J., & Meyerson, M. (2001). Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences*, *98*(24), 13790–13795.

Bilen, M., Işik, A. H., & Yiğit, T. (2020). A new hybrid and ensemble gene selection approach with an enhanced genetic algorithm for classification of microarray gene expression values on leukemia cancer. *International Journal of Computational Intelligence Systems*, *13*, 1554–1566.

Blum, C., Puchinger, J., Raidl, G. R., & Roli, A. (2011). Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing*, *11*(6), 4135–4151.

Bommert, A., Welchowski, T., Schmid, M., & Rahnenführer, J. (2022). Benchmark of filter methods for feature selection in high-dimensional gene expression survival data. *Briefings in Bioinformatics*, *23*, 1–13.

Boughaci, D., & Alkhawaldeh, A. A. s. (2018). Three local search-based methods for feature selection in credit scoring. *Vietnam Journal of Computer Science*, *5*(2), 107–121.

Burhan, M., Chua, K. J. E., & Ng, K. C. (2016). Sunlight to hydrogen conversion: Design optimization and energy management of concentrated photovoltaic (CPV-Hydrogen) system using micro genetic algorithm. *Energy*, *99*, 115–128.

Chakravarty, S., & Mittra, R. (2002). Application of the micro-genetic algorithm to the design of spatial filters with frequency-selective surfaces embedded in dielectric media. *IEEE Transactions on Electromagnetic Compatibility*, *44*(2), 338–346.

Chen, Y., & Song, Z.-J. (2012). Spatial analysis for functional region of suburban–rural area using micro genetic algorithm with variable population size. *Expert Systems with Applications*, *39*(7), 6469–6475.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27.

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis*, *1*(1), 131–156.

Deb, K. (2000). Introduction to selection. *Evolutionary Computation*, *1*, 166–171.

Dif, N., Attaoui, M. w., & Elberrichi, Z. (2018). Gene selection for microarray data classification using hybrid meta-heuristics. In *Modelling and implementation of complex systems* (pp. 119–132). Springer International Publishing.

Dokken, C. B., & Fronk, B. M. (2018). Optimization of 3D printed liquid cooled heat sink designs using a micro-genetic algorithm with bit array representation. *Applied Thermal Engineering*, *143*, 316–325.

Dussaut, J. S., Vidal, P. J., Ponzoni, I., & Olivera, A. C. (2018). Comparing multiobjective evolutionary algorithms for cancer data microarray feature selection. In *2018 IEEE congress on evolutionary computation (CEC)* (pp. 1–8).

Eshelman, L. J. (1991). The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In *Foundations of genetic algorithms* (pp. 265–283). Elsevier.

Gibbons, J. D., & Chakraborti, S. (2020). *Nonparametric statistical inference*. CRC Press.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., & Lander, E. S. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, *286*(5439), 531–537.

Hansen, P., & Mladenović, N. (2018). Variable neighborhood search. In *Handbook of heuristics* (pp. 759–787). Springer International Publishing.

Holland, J. H. (1992). Genetic algorithms. *Scientific American*, *267*(1), 66–73.

Jadhav, S., He, H., & Jenkins, K. (2018). Information gain directed genetic algorithm wrapper feature selection for credit rating. *Applied Soft Computing*, *69*, 541–553.

Jaen-Cuellar, A. Y., Morales-Velazquez, L., de Jesus Romero-Troncoso, R., Moriñigo-Sotelo, D., & Osornio-Rios, R. A. (2016). Micro-genetic algorithms for detecting and classifying electric power disturbances. *Neural Computing & Applications*, *28*(1), 379–392.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *Resampling methods*. New York: Springer (pp. 175–201). Chapter 5.

Jona, J., & Nagaveni, N. (2014). Ant-cuckoo colony optimization for feature selection in digital mammogram. *Pakistan Journal of Biological Sciences*, *17*(2), 266–271.

Ke, W., Wu, C., Wu, Y., & Xiong, N. N. (2018). A new filter feature selection based on criteria fusion for gene microarray data. *IEEE Access*, *6*, 61065–61076.

Kennedy, J., & Eberhart, R. (1997). A discrete binary version of the particle swarm algorithm. In *1997 IEEE international conference on systems, man, and cybernetics, computational cybernetics and simulation* (pp. 4104–4108). IEEE.

Khan, J., Wei, J. S., Ringnér, M., Saal, L. H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C. R., Peterson, C., & Meltzer, P. S. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, *7*(6), 673–679.

Krishnakumar, K. (1990). Micro-genetic algorithms for stationary and non-stationary function optimization. In RodriguezG. (Ed.), *Intelligent control and adaptive systems: Vol. 1196* (pp. 289–296). International Society for Optics and Photonics, SPIE.

Lee, D.-C., Lee, K.-J., & Kim, C.-W. (2019). Optimization of a lithium-ion battery for maximization of energy density with design of experiments and micro-genetic algorithm. *International Journal of Precision Engineering and Manufacturing-Green Technology*, *7*(4), 829–836.

Lee, J., & Kim, D.-W. (2015). Memetic feature selection algorithm for multi-label classification. *Information Sciences*, *293*, 80–96.

Lee, S., Soak, S., Oh, S., Pedrycz, W., & Jeon, M. (2008). Modified binary particle swarm optimization. *Progress in Natural Science*, *18*(9), 1161–1166.

Li, M., Min, Q., Banton, M. C., & Dun, X. (2021). Single-cell regulatory network inference and clustering identifies cell-type specific expression pattern of transcription factors in mouse sciatic nerve. *Frontiers in Cellular Neuroscience*, *15*, 476.

Li, Y., Wang, G., Chen, H., Shi, L., & Qin, L. (2013). An ant colony optimization based dimension reduction method for high-dimensional datasets. *Journal of Bionics Engineering*, *10*(2), 231–241.

Liu, H., & Zhao, Z. (2009). Manipulating data and dimension reduction methods: Feature selection. In *Encyclopedia of complexity and systems science* (pp. 5348–5359). Springer.

Ma, B., & Xia, Y. (2017). A tribe competition-based genetic algorithm for feature selection in pattern classification. *Applied Soft Computing*, *58*, 328–338.

Madadi, R. R., & Balaji, C. (2008). Optimization of the location of multiple discrete heat sources in a ventilated cavity using artificial neural networks and micro genetic algorithm. *International Journal of Heat and Mass Transfer*, *51*(9), 2299–2312.

Mao, Q., & Tsang, I. W.-H. (2013). A feature selection method for multivariate performance measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(9), 2051–2063.

Mendoza, J. E., Morales, D. A., Lopez, R. A., Lopez, E. A., Vannier, J., & Coello Coello, C. A. (2007). Multiobjective location of automatic voltage regulators in a radial distribution network using a micro genetic algorithm. *IEEE Transactions on Power Systems*, *22*(1), 404–412.

Min, F., Hu, Q., & Zhu, W. (2014). Feature selection with test cost constraint. *International Journal of Approximate Reasoning*, *55*(1, Part 2), 167–179.

Mirjalili, S. (2019). Genetic algorithm. In *Evolutionary algorithms and neural networks: Theory and applications* (pp. 43–55). Springer International Publishing.

Moscato, P., & Cotta, C. (2018). An accelerated introduction to memetic algorithms. In *Handbook of metaheuristics* (pp. 275–309). Springer International Publishing.

Moscato, P., et al. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech concurrent computation program, C3P Report, 826:1989.

Moslehi, F., & Haeri, A. (2019). A novel hybrid wrapper–filter approach based on genetic algorithm, particle swarm optimization for feature subset selection. *Journal of Ambient Intelligence and Humanized Computing*, *11*(3), 1105–1127.

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. *Journal of Chemometrics*, *18*(6), 275–285.

Narendra, P. M., & Fukunaga, K. (1977). A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers*, *C-26*(9), 917–922.

Noble, W. S. (2006). What is a support vector machine? *Nature Biotechnology*, *24*(12), 1565–1567.

Oliveto, P. S., He, J., & Yao, X. (2007). Time complexity of evolutionary algorithms for combinatorial optimization: A decade of results. *International Journal of Automation and Computing*, *4*(3), 281–293.

Onan, A. (2018). Biomedical text categorization based on ensemble pruning and optimized topic modelling. *Computational & Mathematical Methods in Medicine*, *2018*.

Onan, A. (2019). Consensus clustering-based undersampling approach to imbalanced learning. *Scientific Programming*, *2019*.

Onan, A. (2022). Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification. *Journal of King Saud University: Computer and Information Sciences*, *34*, 2098–2117.

Onan, A., & KorukoGlu, S. (2015). A feature selection model based on genetic rank aggregation for text sentiment classification. *Journal of Information Science*, *43*, 25–38.

Petricoin, E. F., Ardekani, A. M., Hitt, B. A., Levine, P. J., Fusaro, V. A., Steinberg, S. M., Mills, G. B., Simone, C., Fishman, D. A., Kohn, E. C., & Liotta, L. A. (2002). Use of proteomic patterns in serum to identify ovarian cancer. *The Lancet*, *359*(9306), 572–577.

Pragadeesh, C., Jeyaraj, R., Siranjeevi, K., Abishek, R., & Jeyakumar, G. (2019). Hybrid feature selection using micro genetic algorithm on microarray gene expression data. *Journal of Intelligent & Fuzzy Systems*, *36*(3), 2241–2246.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106.

Rao, H., Shi, X., Rodrigue, A. K., Feng, J., Xia, Y., Elhoseny, M., Yuan, X., & Gu, L. (2019). Feature selection based on artificial bee colony and gradient boosting decision tree. *Applied Soft Computing*, *74*, 634–642.

Rathee, S., & Ratnoo, S. (2020). Feature selection using multi-objective CHC genetic algorithm. *Procedia Computer Science*, *167*, 1656–1664.

Ribas, P. C., Yamamoto, L., Polli, H. L., Arruda, L., & Neves-Jr, F. (2013). A micro-genetic algorithm for multi-objective scheduling of a real world pipeline network. *Engineering Applications of Artificial Intelligence*, *26*(1), 302–313.

Rojas, M. G., Olivera, A. C., Carballido, J. A., & Vidal, P. J. (2020). A memetic cellular genetic algorithm for cancer data microarray feature selection. *IEEE Latin America Transactions*, *18*(11), 1874–1883.

Rostami, M., Berahmand, K., Nasiri, E., & Forouzandeh, S. (2021). Review of swarm intelligence-based feature selection methods. *Engineering Applications of Artificial Intelligence*, *100*, Article 104210.

Santiago, A., Dorronsoro, B., Fraire, H. J., & Ruiz, P. (2021). Micro-genetic algorithm with fuzzy selection of operators for multi-objective optimization: μfame. *Swarm and Evolutionary Computation*, *61*, Article 100818.

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, *2*, 1–21.

Sayed, G. I., Tharwat, A., & Hassanien, A. E. (2018). Chaotic dragonfly algorithm: An improved metaheuristic algorithm for feature selection. *Applied Intelligence*, *49*(1), 188–205.

Sharma, A., & Rani, R. (2019). C-HMOSHSSA: Gene selection for cancer classification using multi-objective meta-heuristic and machine learning methods. *Computer Methods and Programs in Biomedicine*, *178*, 219–235.

Shukla, A. K. (2019). Multi-population adaptive genetic algorithm for selection of microarray biomarkers. *Neural Computing & Applications*, *32*(15), 11897–11918.

Shukla, A. K., Tripathi, D., Reddy, B. R., & Chandramohan, D. (2020). A study on metaheuristics approaches for gene selection in microarray data: Algorithms, applications and open challenges. *Evolutionary Intelligence*, *13*(3), 309–329.

Siedlecki, W., & Sklansky, J. (1993). On automatic feature selection. Chapter 1.3 (pp. 63–87). World Scientific Publishing Co., Inc.

Singh, D., Febbo, P. G., Ross, K., Jackson, D. G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A. A., D'Amico, A. V., Richie, J. P., Lander, E. S., Loda, M., Kantoff, P. W., Golub, T. R., & Sellers, W. R. (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, *1*(2), 203–209.

Sivanandam, S., & Deepa, S. (2008). Terminologies and operators of GA. In *Introduction to genetic algorithms* (pp. 39–81). Berlin Heidelberg: Springer.

Szőllős, A., Šmíd, M., & Hájek, J. (2009). Aerodynamic optimization via multi-objective micro-genetic algorithm with range adaptation, knowledge-based reinitialization, crowding and ϵ-dominance. *Advances in Engineering Software*, *40*(6), 419–430.

Tarle, B., Chintakindi, S., & Jena, S. (2019). Integrating multiple methods to enhance medical data classification. *Evolving Systems*, *11*(1), 133–142.

Truong, V.-H., Nguyen, P.-C., & Kim, S.-E. (2017). An efficient method for optimizing space steel frames with semi-rigid joints using practical advanced analysis and the micro-genetic algorithm. *Journal of Constructional Steel Research*, *128*, 416–427.

Tubishat, M., Idris, N., Shuib, L., Abushariah, M. A., & Mirjalili, S. (2020). Improved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection. *Expert Systems with Applications*, *145*, 113–122.

van't Veer, L. J., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A. M., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., Schreiber, G. J., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R., & Friend, S. H. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, *415*(6871), 530–536.

Vapnik, V. N. (2000). *The nature of statistical learning theory*. New York: Springer.

Venkatesh, B., & Anuradha, J. (2019). A review of feature selection and its methods. *Cybernetics and Information Technologies*, *19*(1), 3–26.

Vijayanand, R., Devaraj, D., & Kannapiran, B. (2018). Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection. *Computers & Security*, *77*, 304–314.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z.-H., Steinbach, M., Hand, D. J., & Steinberg, D. (2007). Top 10 algorithms in data mining. *Knowledge and Information Systems*, *14*(1), 1–37.

Xiong, M., Li, W., Zhao, J., Jin, L., & Boerwinkle, E. (2001). Feature (gene) selection in gene expression-based tumor classification. *Molecular Genetics and Metabolism*, *73*(3), 239–247.

Xue, B., Zhang, M., Browne, W. N., & Yao, X. (2016). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, *20*(4), 606–626.

Xue, Y., & Qin, J. (2022). Partial connection based on channel attention for differentiable neural architecture search. *IEEE Transactions on Industrial Informatics*, 1–10.

Xue, Y., Tong, Y., & Neri, F. (2022). An ensemble of differential evolution and Adam for training feed-forward neural networks. *Information Sciences*, *608*, 453–471.

Yang, C., Chuang, L., Chen, Y., & Yang, C. (2008). Feature selection using memetic algorithms. In *2008 third international conference on convergence and hybrid information technology: Vol. 1* (pp. 416–423).

Yu, L. (2007). Feature selection for genomic data analysis. In *Computational methods of feature selection* (pp. 353–370). Chapman and Hall/CRC. Chapter 17.